Chapter 4
Implementation

4.1 Procedure of speech recognition process

Speech recognition is mainly done in two stages named as training and testing. But before these, some basic techniques that are necessary are applied to these speech signals.

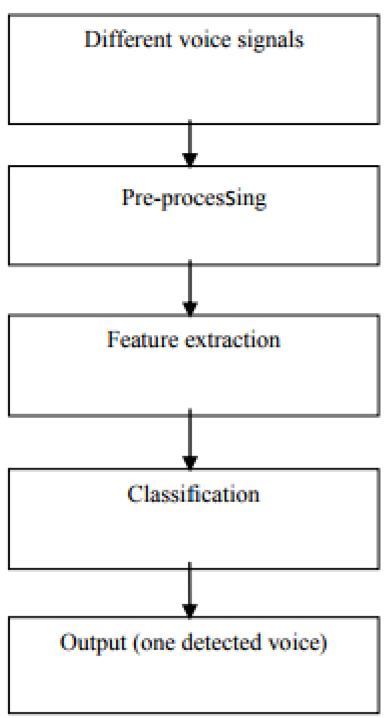


Fig. 4.1 Speech Recognition Process

In this process the voice of different persons is recorded by a good quality microphone in such an environment where no noise is present. These speech signals are then pre- processed by using suitable techniques like filtering, entropy based end point detection and Mel Frequency Cepstrum Coefficient etc. this type of technique makes the speech signal smoother and helps us in extracting only the required signal that is free of noise. Samples are recorded with a microphone. Besides speech signals, they contain a lots of distortion and noise because of the quality of microphone. First of all low and high frequency noise is eliminated by performing some digital filtering.

4.1.1 Speech Classification

Classification of speech signal is very important phenomenon in speech recognition process. Different models are introduced by different authors to classify the speech. But in this work of project, neural network is to be used for classification. A neural network consists of small no of neurons. A Number of neurons are interconnected. A Number of processing units which are used for the processing of speech signals. The very simple techniques like pre-processing, filtering are processed by these types of units. A non-linear weight is computed simply by each unit and the result over its outgoing connection to other units is broadcast. Learning is a process in which value of the appropriate weights is settled. It is necessary whenever we don't have the complete information about the input and output signal. The weights are adjusted by the proposed algorithm to match the input and output characteristics of a network with the desired characteristics. The desired response has to be assumed by our self with the help of teacher. In this work of project, features of the pre- processed speech signals are extracted by using MFCC, LPC. This is called training. The networks are usually trained to perform tasks such as pattern recognition, decision making, and motoric control. Training of the unit is accomplished for the adjustment of the weights and threshold for the classification SVM classifier is used. The feature extraction may be of two types as temporal analysis and spectral analysis. In temporal analysis, the wave formats of the speech signals are analyzed by it. In spectral analysis, the wave format of the speech signal is analyzed by the spectral representation. Except all this, there are some other tools that are necessary to study out are linear predictive coding (LPC) and Mel Frequency Cestrum Coefficients (MFCC). LINEAR Predictive Coding is a tool that is used for the processing of audio signal and speech for representation of spectral envelope and digital signal in compressed form. LPC is based on the idea that expression of each sample of signal in a linear combination of the previous samples. Mel frequency cestrum coefficient is preferred to extract the feature of speech signal. It transforms the speech signal into frequency domain, hence training vectors are generated by it. Another reason of using this method is that human learning is based on frequency analysis. Before obtaining the MFCC of a speech signal the pre emphasis filtering is applied to the signal with finite impulse response filter given by

$(Z)= aprenk$=0(K) $Z{-}K$

Its Z-Transform is $Hpre$=1+ $aprez{-}k$

The value of $apre$is usually taken between -1.0 to 0.4.

Testing is the process, in which different speech signals are tested by using special type of neural network. This is the main step in the speech recognition process. Testing of the speech signals is done after training.

4.1.2 Speech Recognition Process

Recognition of speech is more difficult than the recognition of the printed versions. Various techniques are to be used for the recognition of speech. Basic procedure is shown by the block diagram. It is shown that how speech can be recognized using different processes.

Speech is used effortlessly by humans as a mode of communication with one another. Same type of easy and natural communication is wanted with machines by people. So, speech is preferred as an interface rather than using any other interfaces like mouse and keyboard. The speech recognition process is somewhere difficult and complicated phenomenon. The speech recognition system can further be divided into various classes. It may be classified based on the model of speaker and type of vocabulary.

This figure shows the general procedure of the speech recognition process. Typical speech sentence consist of two main parts; speech information is carried out by one part and silent and noise sections between the utterances without any verbal information is carried out by the other part. At the input side, different voice signals are applied. Before applying these signals to the neural network, pre- processing of the signals is done by using filtering; Entropy based end point detection and MFCC. The audio signals are converted into particular waveforms. The next step is to extract the features of the voice signals by the special kind of neural network. Neural Network acts as the brain of human. Trained neural networks trains these networks and at the last testing of voice signals is done. Tested signal is detected as the output. All the working procedure is shown in the block diagram of speech recognition process how the steps take place.

4.1.3 Voice Individuality

Before trying to solve the problem described in the goal of project, we must understand the characteristics of the different voice signals. Acoustic parameters have the greatest influence on the voice individuality. Acoustic parameters may be divided into two types: time dimensions that represent the pitch frequency or fundamental frequency and in frequency dimensions that represent the vocal tract resonance. We can consider the voice signals as quasi periodic signals. Pitch may be defined as the fundamental frequency of the voice signal. The average pitch speed, time pattern, gain and fluctuation change from one individual to another and also within the speech of the same speaker in actual the frequency response of the vocal tract filter is the shape and gain of the spectral envelope of the signal. From some researches on voice individuality, it has been concluded that pitch fluctuation that gives the second place to the format frequencies is the most important factor in the voice individuality. From many other studies it also be concluded that the spectral envelope has the greatest influence on the voice individuality perception.
From the above discussion it is concluded that there is no single parameter that can alone define a speaker. A group of parameters that depend on the nature of speech materials vary from one individual to another having their respective importance.

4.2. Database Generation

1. Load ‘normlpcdatabase.mat’
2. Run the normalized pc file
3. Recorded the numbers and named them accordingly
4. Each individual spoke 0 to 9 digits thrice
5. Database of 5 individuals are recorded
6. In all, 150 samples are recorded
7. Samples were trained using neural networks

4.3. Conversion of speech signals into waves

The samples of the speech signals are converted into wave formats. This is the most general way for the representation of the signal. A disadvantage of this method is also there which is that it cannot represent speech related information. This problem may be solved by the technique pre-processing. This representation shows the change in amplitude spectra over time. There are three dimensions. X-axis represents time in meter per second. Y-axis represents frequency and z-axis represents the color intensity that represents the magnitude of the signal. It is not possible to start samples exactly at the same time because different persons pronounce sentences differently i.e. slowly or fast and as the result intensities at the different times might be different.

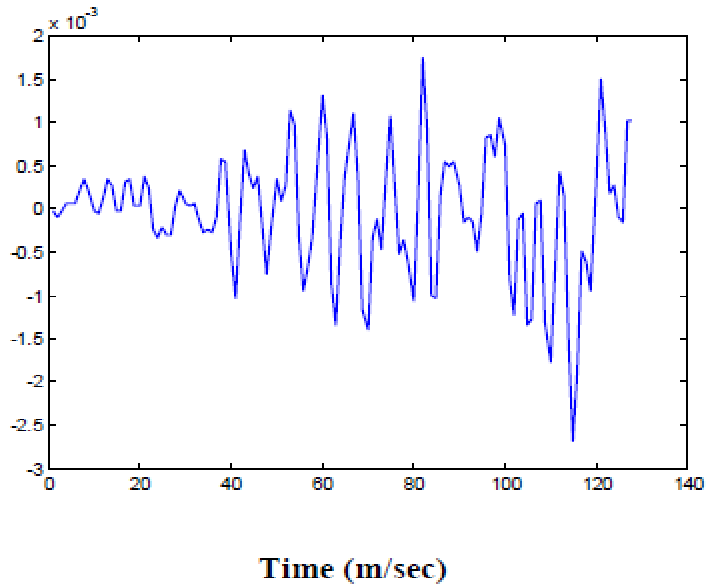


Fig.4.3 Wave Format of a Speech Signal

This figure shows the wave format of a speech signal obtained in the implementation in the MATLAB. The change in amplitude spectra over time is shown by the domain representation. The complete sample is split into two time frames with almost 50% overlap. We calculate the short term frequency for each time. Although good visual representation of speech signal is provided by the spectrogram, verification between samples is still there. Samples never start exactly at the same time, sentences are pronounced differently by different persons slower or faster and as a result that might have different intensities at different times.

4.3.1 Algorithm Used

Here the algorithm used is the back propagation. The back propagation algorithm was originally introduced in the 1970s. Several neural networks are described here, in which back propagation works faster than the earlier approaches used for the learning phenomenon. It makes possible to use neural network to solve problems which have not been solved for a long time. In today's world, the back propagation algorithm is the work horse of learning in neural network. Besides this, it gives us detailed insights into how changing the weights and how the overall behavior of a network is changed by the biases.

4.3.2 Role played by Neural Network in Speech Recognition Process

Neural network works as a human brain. These networks perform learning phenomenon. Neural network is a computational model inspired by an annual central nervous system which is capable of machine learning as well as pattern recognition the artificial neural networks are generally presented as systems in which number of neurons are interconnected which have been used to solve a wide variety of tasks that are difficult to solve using ordinary rule based programming including computer vision, speech recognition. This type of network potentially contains a large number of simple processing units, roughly analogous to neurons in the brain. All these units are operated simultaneously. Except neural network, there is no other processor that oversees their activity. These units perform all computations in the system. A scalar function is computed simply by each unit and the result is broadcasted to its neighboring units. The course of dimensionality problem that many attemptations to model non- linear functions with large number of variables is also kept in check by neural network. Representative data is collected by neural network users and then training algorithms are invoked because they have to learn the structure of the data automatically. One method for estimating a continuous target for training patterns of neural networks that are based on the generalized regression neural network and they compared the performance with the performance of linear and multilayer perception.

There are two input units in a network by which data is received from the environment, hidden layers y which transformation of data is represented internally output units whose function is to take decision. It is possible train recurrent neural networks for sequence labeling problems where the input and output alignment is not known by end to end training method such as connectionist temporal classification. Hence neural network plays a great role in recognizing the speech in this work of project.

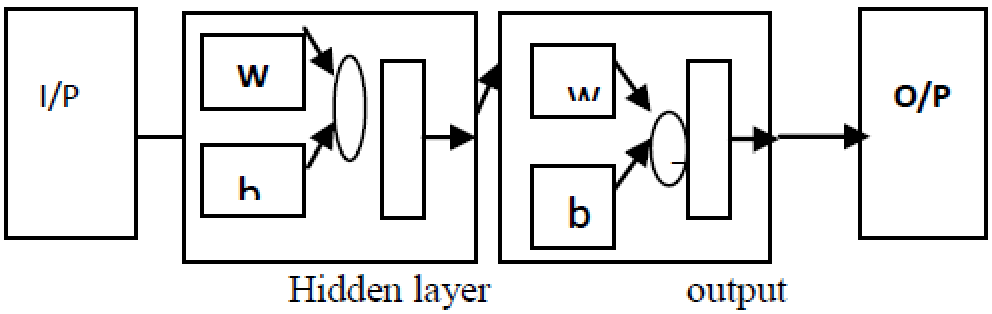


Fig.4.3.2 Basic Structure of Neural Network

4.4 Setting of ZigBee Network

ZigBee is a protocol that uses the 802.15.4 standard.

ZigBee defines three different device types:

- Coordinator
- Router
- end devices

4.4.1. Coordinator

- A coordinator has the following characteristics:
- Selects a channel and PAN ID (both 64-bit and 16-bit) to start the network
- Can allow routers and end devices to join the network
- Can assist in routing data
- Cannot sleep--should be mains powered.

4.4.2. Router

A router has the following characteristics:

- Must join a ZigBee PAN before it can transmit, receive, or route data
- After joining, can allow routers and end devices to join the network
- After joining, can assist in routing data
- Cannot sleep should mains powered

4.4.3. End Devices

An end device has the following characteristics:

- Must join a ZigBee PAN before it can transmit or receive data

- Cannot allow devices to join the network
- Must always transmit and receive RF data through its parent. Cannot route data.
- Can enter low power modes to conserve power and can be battery-powered

ZigBee networks are called personal area networks or PANs. Each network is defined with a unique PAN identifier (PAN ID). If multiple ZigBee networks are operating within range of each other, each should have unique PAN IDs. In ZigBee networks, the coordinator must select a PAN ID and channel to start a network. Every ZigBee has other critical parameters such as Destination Address High, Destination Address Low; these parameters have to be the same across all devices to be configured.

4.4.4. ZigBee Transreceiver

ZigBee is an IEEE 802.15.4based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection.

Fig. 4.4.4 ZigBee Module

4.4.4.1 Characteristics of ZigBee

- ZigBee is a low power network.
- ZigBee is a low data rate network.
- ZigBee is a close proximity (i.e., personal area) wireless ad hoc network.
- ZigBee technology less expensive.
- Transmission distance is 10–100 meters (line-of-sight).
- ZigBee is typically used in low data rate applications that require long battery life and secure networking.
- ZigBee has a defined rate of 250 Kbit/s.

4.4.4.2. ZigBee Configuration

4.4.4.2.1 Brief introduction, Download and install XCTU

This section contains download and install instructions based on operating system. XCTU is compatible with Linux, OSX, and Windows. It may be necessary to configure your system prior to installing XCTU for the first time.

Operating systems:

XCTU is compatible with the following operating systems:

- Windows Vista/7/8/10 (32-bit or 64-bit versions)
- Mac OS X v10.6 and higher versions (64-bit only)
- Linux with KDE or GNOME window managers (32-bit or 64-bit versions)

4.4.4.2.1.1 Supported RF modules

XCTU supports configuration and communication for most Digi RF modules. XCTU uses a serial link to interact with these radio modules, providing an easy-to-use and intuitive graphical interface. The following is a complete list of XCTU-compatible RF modules:

- Xbee®/Xbee-PRO® RF Module Family
    - Xbee SX
    - Xbee-PRO SX
    - Xbee 802.15.4
    - Xbee-PRO 802.15.4
    - Xbee ZB
    - Xbee-PRO ZB

- Programmable Xbee-PRO ZB
- Xbee ZB SMT
- Xbee-PRO ZB SMT
- Programmable Xbee-PRO ZB SMT
- Xbee-PRO 900HP
- Programmable Xbee-PRO 900HP
- Xbee-PRO XSC
- Xbee-PRO 900
- Xbee-PRO DigiMesh 900
- Xbee DigiMesh 2.4
- Xbee-PRO DigiMesh 2.4
- Xbee-PRO 868
- Xbee Wi-Fi
- Xbee 865LP
- Programmable Xbee 865LP
- Xbee Cellular
- Xbee 868LP SX
- Xbee Thread
- XBee3

- XT end® RF Module family
- XLR PRO radio solution
- n XLR Module

4.4.4.2.1.2 Configuration Mode

- **Configuring in AT mode**
- In AT operating mode, you must put the module in a special mode called command mode so it can
- Receive AT commands. For more information about configuring RF modules working in AT operating mode.
- **Configuring in API mode**
- To configure or execute AT commands when the RF module is in API operating mode, you must

- Generate an AT command API frame containing the AT setting identifier and the value of that setting, and send it to the RF module.

4.4.4.2.1.3 XCTU overview

XCTU is divided into five main sections: the menu bar, main toolbar, devices list, working area, and status bar.

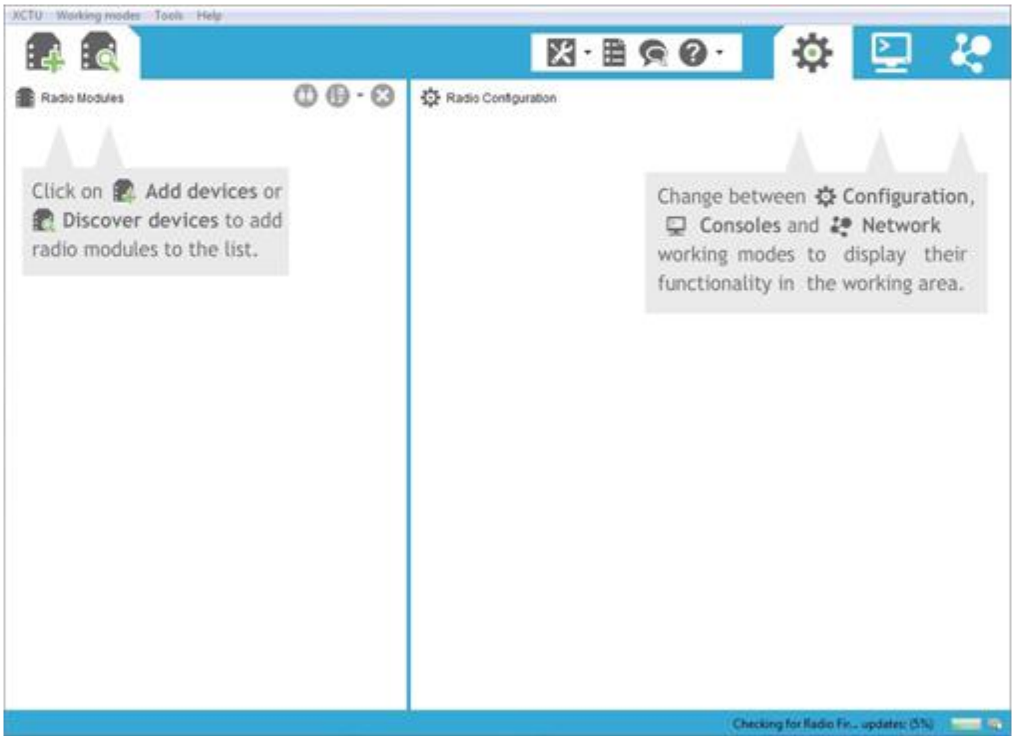


Fig. 4.4.4.2.1.3(a) XCTU window

- **Menu bar**

The menu bar is located at the top of the application. You can use the menu bar to access all XCTU features, tools, and working modes.

Fig. 4.4.4.2.1.3(b) Menu bar

- **Main toolbar**

The main toolbar is located at the top of the application and is divided into three sections.

Fig. 4.4.4.2.1.3(c) Main toolbar

- The first section contains two icons used to add radio modules to the radio modules list.

- The second section contains the static XCTU functionality that does not require a radio module. This section includes the XCTU tools, the XCTU configuration, the feedback form, and the help and updates functions.

- The third section contains tabs corresponding to the three XCTU working modes. To use this functionality, you must have added one or more radio modules to the list.

  - **Devices list**

The radio modules list, or devices list, is located on the left side of the tool and displays the radio modules that are connected to your computer. If you know the serial port configuration of a radio module, you can add it to the list directly. You can also use the discovery feature of XCTU to find radio modules connected to your PC and add them to the list. Depending on the protocol of the local radio modules added, you can also add remote radio modules to the list using the module's search feature.

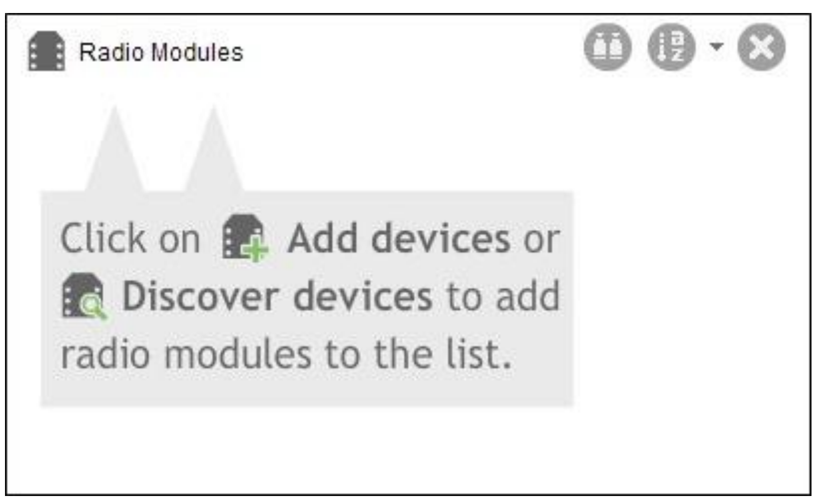


Fig. 4.4.4.2.1.3(d) Search module

- **Working area**

The working area is the largest section and is located at the right side of the application. The contents of the working area depend on the working mode selected in the toolbar. To interact with the controls displayed in the working area, you must have added one or more radio modules to the list and one of the modules must be selected.

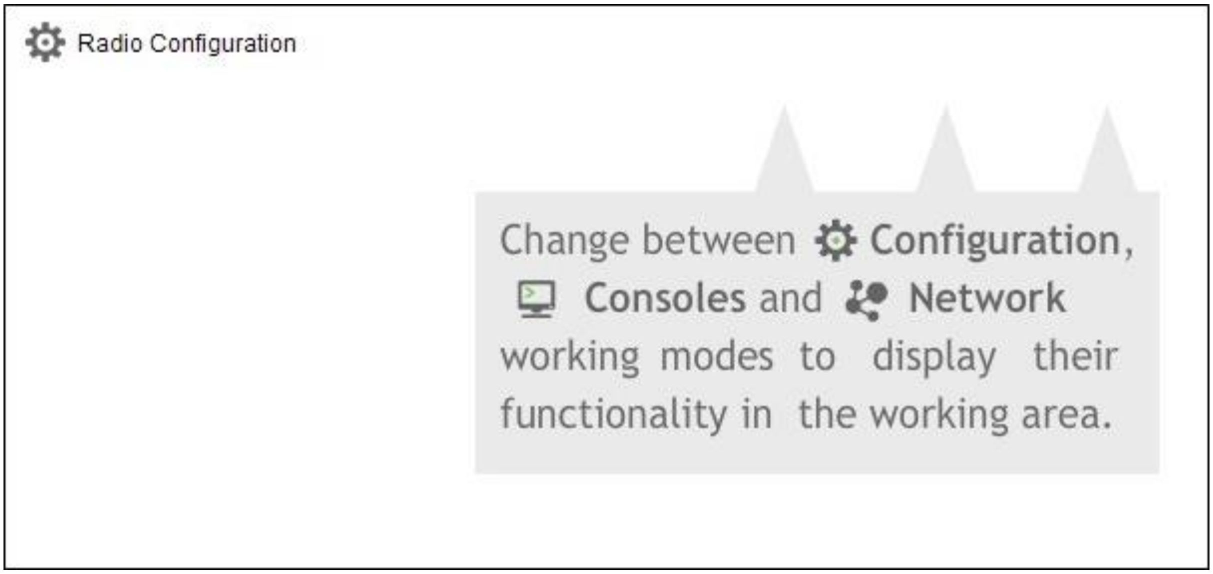


Fig. 4.4.4.2.1.3(e) Functionality of working area

- **Status bar**

The status bar is located at the bottom of the application and displays the status of specific tasks, such as the firmware download process.

Fig. 4.4.4.2.1.3(f) Status bar

4.4.4.2.1.4 Add a radio module manually

If you know the serial configuration of your radio module, you can add it to the list manually.

1. Click the **Add a radio module** button   from the toolbar. The **Add a radio module** dialog opens.

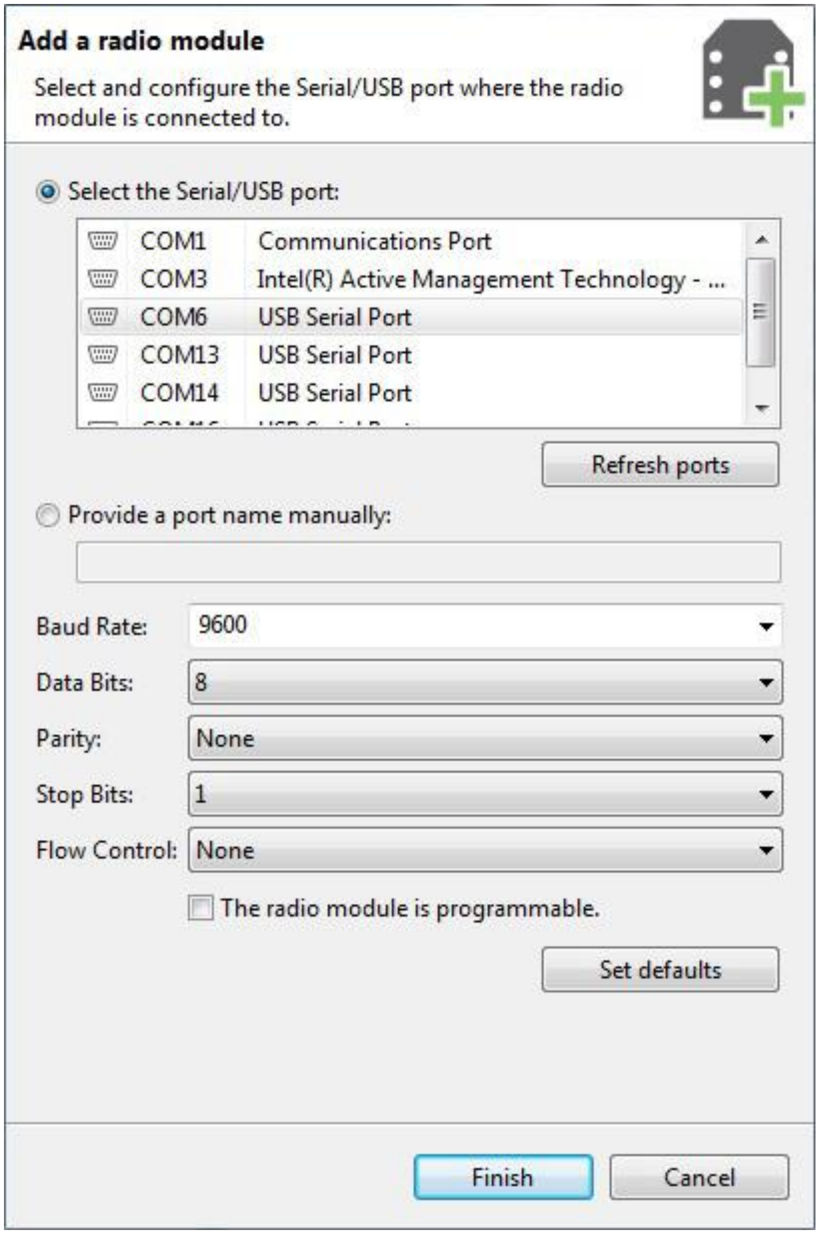


Fig. 4.4.4.2.1.4(a) Port selection window_1

2. Select the serial port where the radio module is connected (or enter its name manually) and configure the serial settings of the port.
3. Click **Finish** to add the radio module to the list of radio modules.
4. If the settings were configured correctly and the radio module was connected to the selected port, the module is displayed in the device list. For more information about the device list

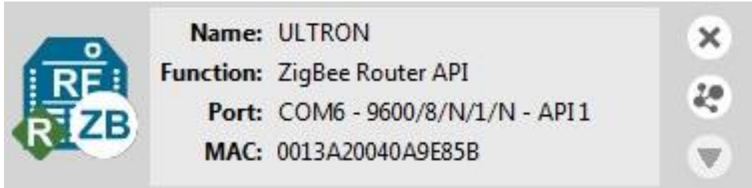


Fig. 4.4.4.2.1.4(b) Selected ZigBee Module_1

5. If the settings were configured incorrectly, an **Action required** dialog asks you to reset the module. Reset the module. The action required dialog should close and your module should be added to the list.
6. If your module could not be found, XCTU displays the **Could not find any radio module** dialog providing possible reasons why the module could not be added.

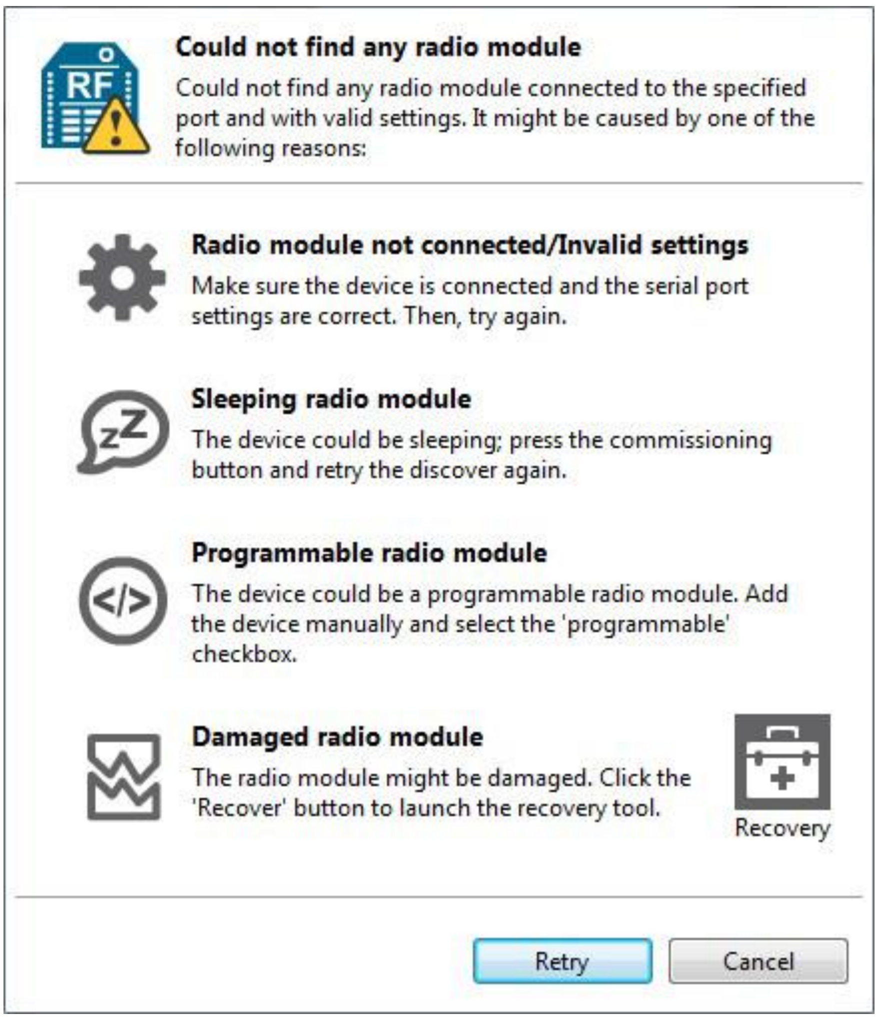
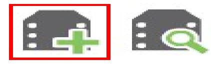


Fig. 4.4.4.2.1.4(c) Error Window

4.4.4.2.1.5 Add a programmable radio module

Some radio module variants are programmable, which means they are able to run applications written in C. Normally, they are known as Programmable Xbee modules and can be identified by a part number ending in B on the back label. To add a programmable radio module:

1. Click the **Add a radio module** button  from the toolbar. The **Add a radio module** dialog opens.

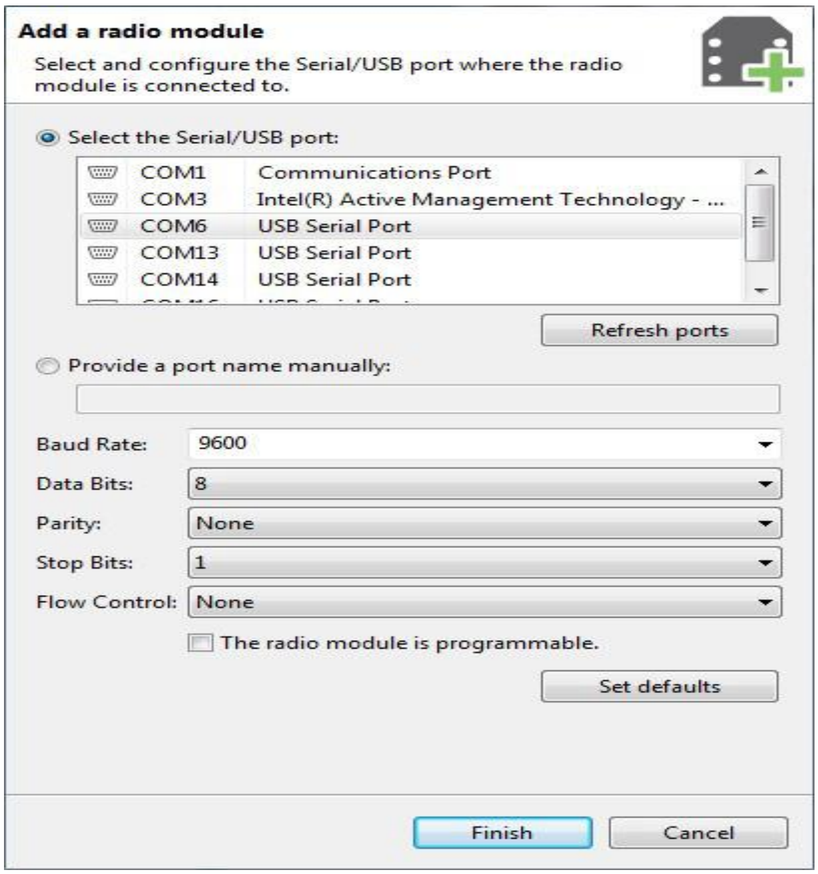


Fig. 4.4.4.2.1.5(a) Port selection window_2

2. Select the serial port to which the radio module is connected (or enter its name manually) and configure the serial settings of the port.
3. Check the **My radio module is programmable** setting.

4. Click **Finish**.

5. Reset your radio module when prompted. The module appears in the device list.

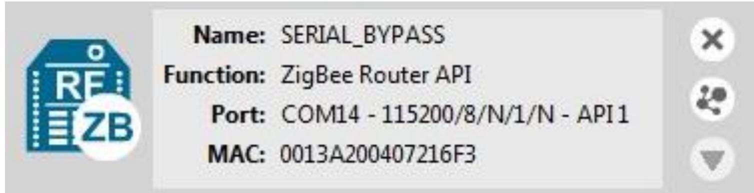


Fig. 4.4.4.2.1.5(b) Selected ZigBee Module_2

4.4.4.2.1.6 Discover local radio modules

XCTU can discover radio modules that are connected directly to your computer. You can use the discovery tool if you don't know the serial configuration of your radio module, don't know the port it is connected to, or want to add multiple modules at once.

1. Click the **Discover radio modules** button  on the XCTU toolbar. The **Discover radio modules** dialog box opens.

Fig. 4.4.4.2.1.6(a) Multiple port selection window

2. Select the serial ports you would like to scan for radio modules. Click **Next**.

3. Select any port parameters you would like to include in the search process.

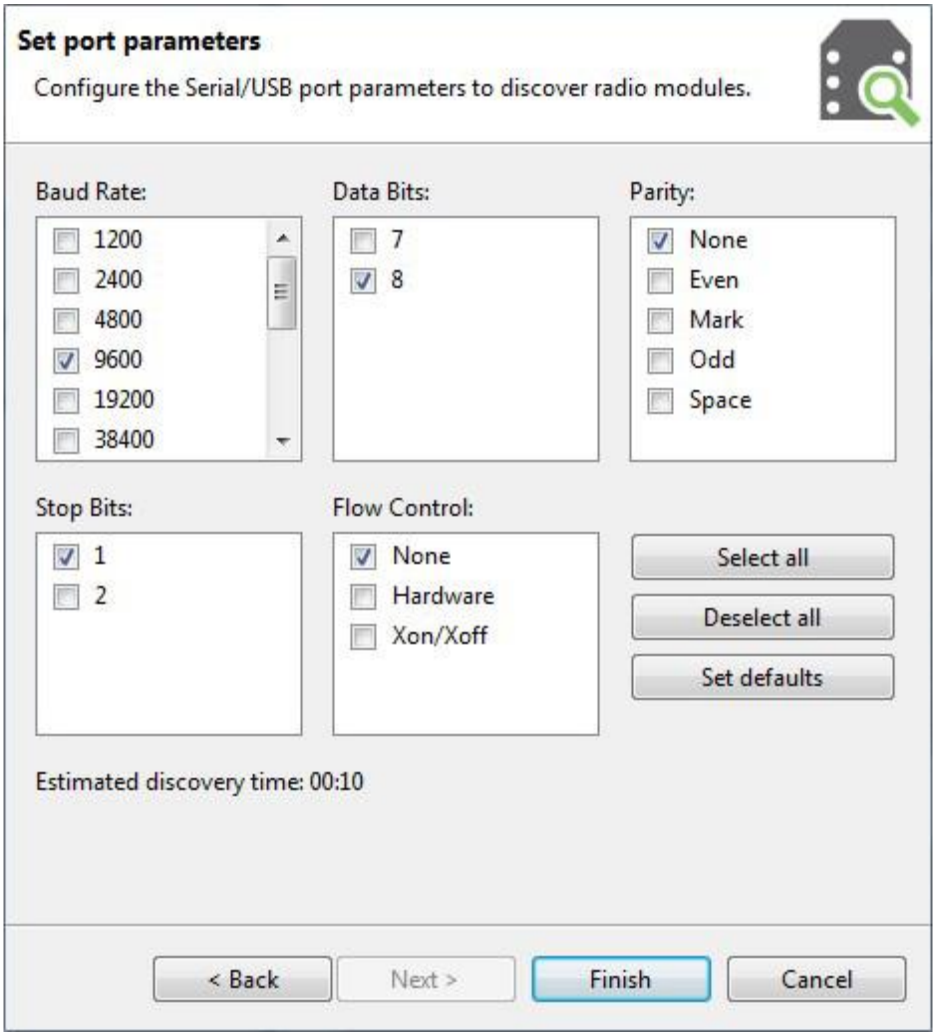


Fig. 4.4.4.2.1.6(b) Setting parameter window

4. Click **Finish** to initiate the discovery scan. A new dialog opens, displaying devices found and estimated time remaining. You can click **Stop** to halt the discovery process at any time. For example, you can stop the process if the modules you were looking for are already found.

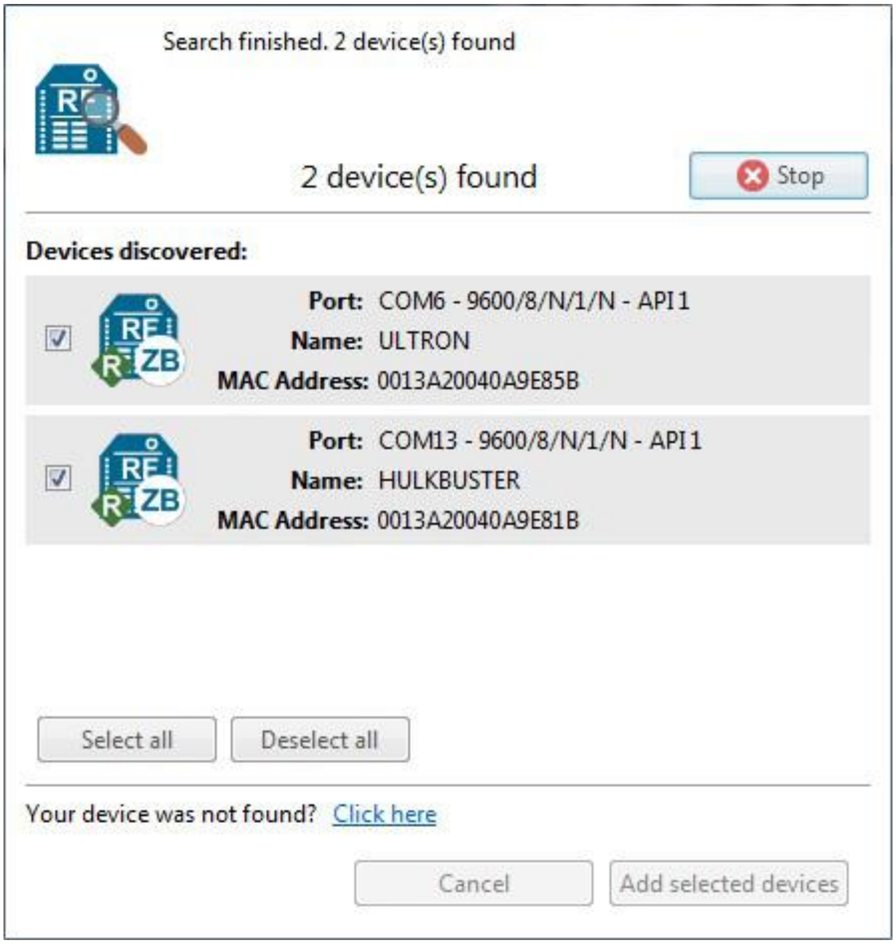


Fig. 4.4.4.2.1.6(c) Multiple ZigBee module selection window

5. Select the box next to the module(s) you want to add to your device list and click **Add selected devices**. The modules appear in the device list.

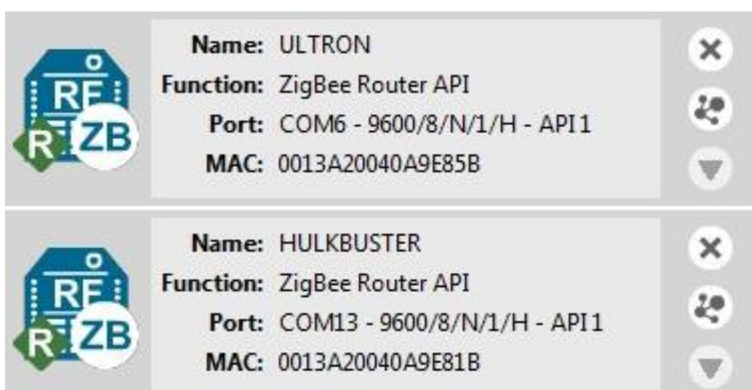


Fig. 4.4.4.2.1.6(d) Selected ZigBee module in API mode

4.4.4.2.1.7 Radio module information panel

Local radio modules appear as big buttons in the modules list. Each module displays identifying information about itself. To work with a radio module, you must select it from the list of devices. When you hover over a module, the background color changes to yellow. Selecting a radio module refreshes the contents of the working area, displaying the information or actions you can perform on the selected module. The contents of the working area depend on the active working mode.

4.4.4.2.1.8 Module information box

When you hover over the icon, XCTU displays additional information about the selected module, including module type, family, protocol, device type, firmware, and hardware.

Module type:	XB24-ZB
Family:	XBEE
Protocol:	ZigBee
Device type:	Router
Firmware:	23A7
Hardware:	0x19

Fig. 4.4.4.2.1.8 Module information Window

4.4.4.2.1.9 Discover remote radio modules

You can execute a discovery process to locate remote radio modules in the same network as the local (selected) module. To discover remote modules:

1. Select a module from your device list. If you do not have any modules in the list.
2. Click the **Discover radio nodes in the same network** button. As XCTU discovers new remote radio modules, they appear in the discovery process dialog box.
3. Click **Stop** to halt the discovery process at any time.
4. Check the box next to the module(s) you want to add to your device list and click **Add selected devices**. The discovered remote modules appear in the list of remote modules.

4.4.4.2.2 Configure modules

You will need to configure your Xbee modules so they can communicate. The first part of this configuration involves setting the Channel, PAN ID, and Address values.

- **Channel**: The channel calibrates the operating frequency within the 2.4GHz 802.15.4 band. Your Xbee's must be on the same channel to communicate with one another.
- **PAN ID** (Personal Area Network ID): Your Xbee's must share the same PAN ID to communicate with one another. You can choose a value between 0 and 0xFFFF.
- **Addressing**: Each Xbee has a source address (referred to as "MY address") and a destination address (which has an upper half, Destination High or DH, and a lower half, Destination Low or DL). An Xbee's destination address specifies to which source address it can send data. You can specify a universally unique address by using the 64-bit address printed on the back of the module, use a shorter 16-bit address (unique within a network), or use a string of text (e.g., "Alice's radio").

Additionally, each Xbee in a network plays a role. The three role options are Coordinator, End Device, and Router. Each network has exactly one Coordinator, which serves as the root of the network tree. A network can have multiple Routers; these can forward information to end devices and also run application functions. Lastly, End Devices cannot relay data, but only talk to a parent node (either a Coordinator or Router). A network can have multiple End Devices.

With that, let's configure our Xbee's.

1. Download and install XCTU. Available for both Windows and Mac.
2. Plug your first Xbee into an Explorer module, and connect to your computer's USB port via a USB cable.
3. Open XCTU and click "Discover devices."

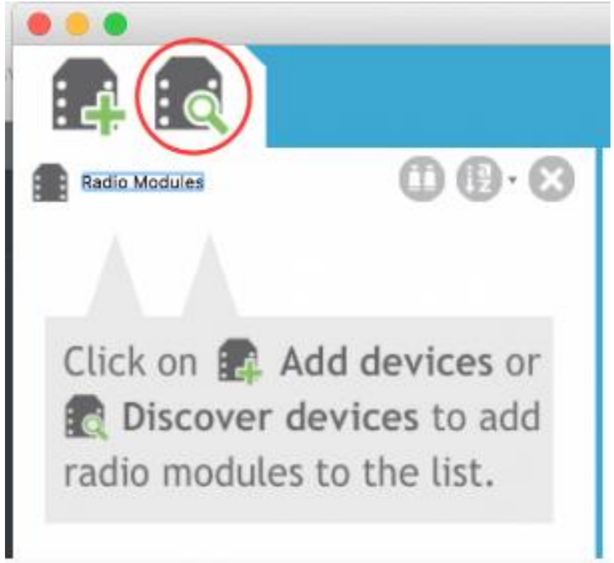


Fig. 4.4.4.2.2(a) Select Add Device window

4. Select the port to be scanned. Then on the next page, select the settings as shown below. Click "Finish".

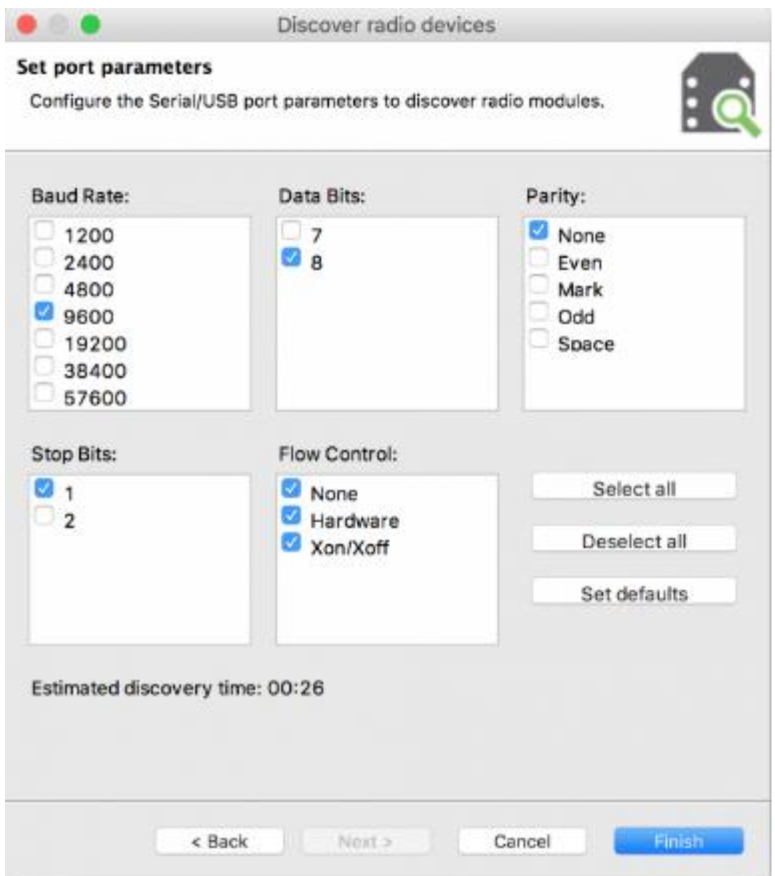


Fig. 4.4.4.2.2(b) Select parameter window

5. Your device should appear on the “Devices discovered” list. Click “Add selected devices” for your module.
6. With the Gear icon selected, click the radio module in the left-hand menu: This should open up a long list of settings.
7. Pick a channel (between 0x0 and 0xFFFF). Remember, each Xbee must share the same channel number. I chose an arbitrary value of “0x0C.”
8. Pick a PAN ID. As before, both modules must have the same number.
9. For this simple demo, I went with the addressing configuration shown below (the 16-bit option from the address options previously discussed).
10. Lastly, select “Coordinator” from the Coordinator Enable (CE) dropdown list. After you’ve made all your changes, click “Write.”
11. Now unplug the first Xbee, and repeat steps 2 through 10 with the second Xbee, but with the addresses as shown in the second column of the table above. On this second Xbee, you can leave the Coordinator Enable (CE) set as 0, or End User. I recommend you mark each module so you can easily tell them apart.

Run a Communication Test

1. Now it’s time to see if they’ll talk. Plug one Xbee into a USB port, and plug the other Xbee into another USB port. In XCTU, scan for devices as before. You should see both devices available. Select both of them and click “Add selected devices.”

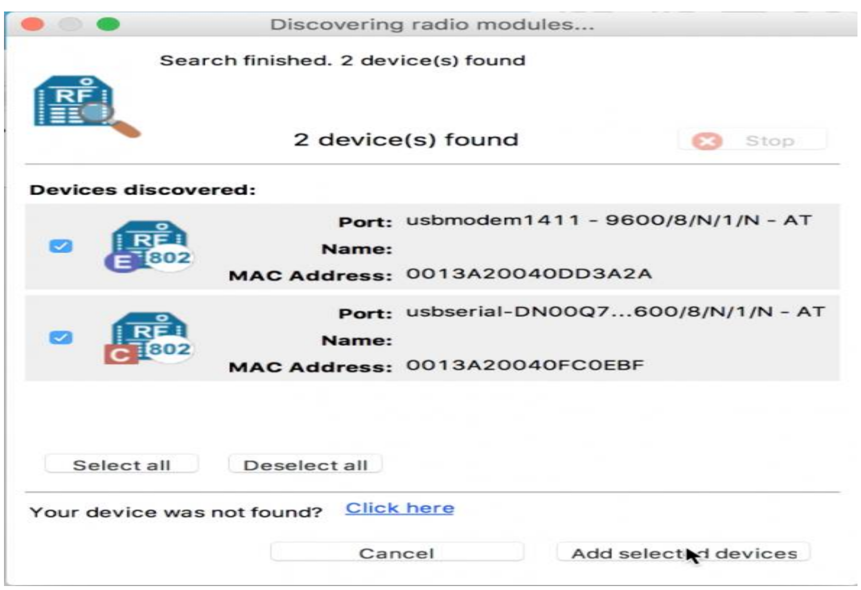


Fig. 4.4.4.2.2(c) ZigBee module selection

2. Click one of the modules in the left-hand column. Now select the Console icon to view the console. Click "Open."

Fig. 4.4.4.2.2(d) Console Window

3. Repeat for the other module, opening up a console.
4. Type into one console. You should see the result echoed back in the other console. If so, congrats!

4.5. Arduino Implementation

Step 1: Install Arduino packages in MATLAB

Step 2: Type ‘Arduino’ command in MATLAB and get information about the following

- Available digital pins in Arduino
- Available analog pins in Arduino
- Comport
- Board type

Step 3: Check whether to which comport Arduino is connected

Step 4: Write the code in Arduino and compile the code

Step 5: Upload the code in Arduino board

Step 6: open Matlab, comport and baudrate in Matlab and Arduino should be same

Step 7: Run the code in Matlab

Step 8: According to the code the DC motor runs